VEHICLE-BASED ORDER ENTRY AND PROCESSING MECHANISM

FIELD OF THE INVENTION

The present invention relates to data processing systems. More particularly, the present invention relates to remote order entry and processing.

BACKGROUND OF THE INVENTION

The phrase "modern conveniences" has been uttered in American society for decades. One of the earliest conveniences was the service concept of "eating out." Here, we mean the notion of purchasing a meal in a restaurant instead of preparing a meal in one's home. The appeal to this particular service is based on several possible advantages, including a specialized restaurant environment, specialized food (e.g., French Cuisine), or simply elimination of the effort associated with preparing a meal.

Other conveniences have been brought on by the introduction of new technologies. For instance, the proliferation of affordable automobiles in the early 1900's brought about a significant change to the American lifestyle. Middle class America became an extremely mobile society, with daily travel of several miles becoming commonplace. By the 1950s, there was a perceived need to integrate the automobile into the restaurant experience. This patent pertains to this automobile-restaurant integration.

Perhaps the earliest attempt at integration was the "car hop" concept. Restaurants were designed to have a kitchen and a large car port. The car port would typically be large enough to accommodate several cars. Customers would enter the

car port in their vehicles and be waited on by waitpersons known as "car hops." While the car hop arrangement provided an enjoyable and often fun experience, it was quite inefficient. The car ports were limited in size, and providing service to each individual car took a fair bit of time. One technologic improvement to the car hop arrangement was the use of "service phones." Service phones were used by customers to place orders directly from their vehicles, meaning that car hops were no longer responsible for taking customer orders, but were instead only responsible for delivering the food needed to fill the orders. However, while service phones helped, the physical limitations of the car port and the effort needed to delivery the food continued to make the car hop arrangement inefficient.

Fast food restaurants came on the scene at about the same time as car hop restaurants. While early fast food restaurants were not a direct attempt to integrate the concept of the automobile with that of the restaurant, the notion of "fast" food did not have much meaning in the pre-automobile era. After all, it did not really matter how "fast" the food could be prepared if transportation to and from the restaurant was impractical. Even still there was the perceived need to better integrate the concepts of the automobile and fast food. A well-known solution to this need for integration is the "drive-up window." While no one can be sure where and when the first drive-up window was used, drive-up windows became fairly common in the 1970's and 1980's; and today, it is difficult to find a fast food restaurant without a drive-up window.

While drive-up windows currently represent the greatest degree of automobile-restaurant integration, present solutions are extremely inefficient. The order processing used in today's drive-up window arrangements is basically the same as that used inside the fast food restaurant itself. Customers wait in line, determine what they want to order, and present their order to the teller. A menu is displayed for review by the customers. The customers then determine what to order and present their order to the teller when asked.

While this line-oriented process works well inside the restaurant, it is exceedingly inefficient outside the restaurant in the drive-up window line. Because of the size of automobiles and the distance between each automobile, only one customer is able to see the menu at a time. Thus, customers loose the opportunity to formulate their order while waiting in line. This problem is exacerbated by the fact that the drive-up menu and the ordering position are typically located at the same place, which means that the customer is asked to formulate their order and present it to the teller at the same time. This recurring scenario causes several problems. First, the customer is frustrated and annoyed because they are being asked to do two things at once. Second, the customer will often need to ask for more time, which costs the restaurant money. Third, individuals in cars behind the ordering car become annoyed and frustrated with the delay and in some cases communicate their frustration to the ordering car, resulting in embarrassment and further annoyance and anger.

Yet another problem with today's drive-up window process is the two-way speaker system that is typically used to aurally exchange orders and information between the automobile and the teller. Often times, traffic and engine noise make communicating a difficult proposition, which of course causes additional frustration and annoyance.

Without an improved mechanism for processing automobile orders, the fast food industry will continue to annoy and frustrate customers and waste valuable time and money.

SUMMARY OF THE INVENTION

The preferred embodiment of the present invention uses an order processing server to transmit an electronic menu to a customer's hand held device or to a computer located within a customer's automobile. The server repeatedly transmits the menu, and when a vehicle comes within range of the server's transceiver, the menu is received by the particular customer device. The order is then formulated and transmitted back to the server.

These and other features of the present invention will be explained in further detail in the text associated with the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of the order processing server that is used in the preferred embodiment of the present invention.

Figure 2 is a block diagram of the customer device that is used in the preferred embodiment of the present invention.

Figure 3 shows the steps used to carry out the processing of menu transmission thread of the preferred embodiment.

Figure 4 shows the steps used to carry out the processing of the Menu Processor of the preferred embodiment.

Figure 5 shows the steps used to carry out the processing of the order processing thread of the preferred embodiment.

Figures 6 and 7 are hierarchical diagrams showing the menu structure of the preferred embodiment.

Figure 8 is a diagram showing a document type definition for the MDML language.

Figures 9 and 10 are example MDML documents that are used in the Detailed Description to help explain the innerworkings of the preferred embodiment.

DETAILED DESCRIPTION

Server 100

Turning now to the drawings, Figure 1 is a block diagram of the server computer system of the preferred embodiment. Server 100 is an enhanced IBM Personal Computer 300PL; however, it should be understood that the present invention is not limited to any one make or type of computer system. As shown, Server 100 comprises Central Processing Unit (CPU) 105, which is connected to Serial Port 110, Display Adapter 120, Auxiliary Storage Adapter 125, and Main Memory 135. These system components are interconnected through the use of System Bus 130. As shown, Serial Port 110 is also connected to Wireless Transceiver 160.

CPU 105 is a 233 MHz. Pentium Processor made by Intel Corporation. However, it should be understood that the present invention is not limited to any one make of processor and that the invention could be practiced using some other type of a processor, such as a co-processor or an auxiliary processor. Auxiliary Storage Adapter 125 is used to connect mass storage devices (such as a Hard Disk Drive) to Server 100.

Main Memory 135 contains Operating System 140, Menu/Price Database 145, XML Parser 150, and Order Processing Server 155. Menu Transmission Thread 157 and Order Processing Thread 159 are different threads running under the task of Order Processing Server 155.

Operating System 140 is Windows NT, which is a well-known multi-tasking and multi-threading operating system offered and sold by Microsoft Corporation. As its name suggests, Menu/Price Database 145 is used to store menu and price information. This information is used by Menu Transmission Thread 157. XML

Parser 150 can be one of any of the available XML Parsers available for the Windows NT Operating System. Order Processing Server 155 and its threads, Menu Transmission Thread 157 and Order Processing Thread 159, are explained using Figures 3 and 5-10 and the associated text.

Server 100 utilizes well-known virtual addressing mechanisms that allow its programs to behave as if they have access to a single, large storage entity (i.e., instead of access to multiple, smaller storage entities such as Main Memory 135 and a HDD). Therefore, while certain mechanisms and constructs are shown to reside in Main Memory 135, those skilled in the art will recognize that these programs are not necessarily all completely contained in Main Memory 135 at the same time. For example, portions of Operating System 140 will reside in Main Memory 135 while executing on CPU 105, but will at other times reside on an attached HDD. Thus, the term *memory* is used herein to generically refer to storage that spans the entire virtual address space of a computer system, irrespective of the particular physical devices that make up that storage.

Display adapter 120 is used to directly connect a display device to computer system 100. Serial Port 110 is used to connect Server 100 to other devices such as Wireless Transceiver 160. Wireless Transceiver 160 is used to continually transmit a menu in XML format as a wireless transmission. This aspect of the preferred embodiment is described in the text associated with Figure 4. The wireless protocol used in the preferred embodiment is that known in the industry as Bluetooth, which is a wireless protocol standard that is being used by various companies within the industry. However, it should be understood that other short-range wireless connectivity standards could be used such as that promulgated by the InfraRed Data Association (IRDA). Another important point to note is that the range of the wireless transmission must be closely tailored to the environment at issue to avoid reception of

the wireless transmission by non-customers. For example the range contemplated in the Bluetooth protocol is ten (10) meters.

Two hardware embodiments of Customer Device 200 will now be presented. The term Customer Device should be understood in the specification and in the claims to refer to either hardware environment and any legal equivalents thereof. For example, Customer Device 200 could be a laptop computer system or be imbedded into any one of a number of portable devices such as cellular phones or other communication devices.

Customer Device 200 - PALM III

The first hardware embodiment of Customer Device 200 is a PALM III Personal Digital Assistant (PDA) device that is marketed by 3COM Incorporated; however, it should be understood that any other similarly configured PDA could be used. As shown, Customer Device 200 comprises Central Processing Unit (CPU) 205, which is connected to Serial Port 210, LCD Controller 220, and Memory Card 235. These system components are interconnected through the use of System Bus 230. As shown, Serial Port 210 is also connected to Wireless Transceiver 260.

CPU 205 is a 68000 series embedded processor that is manufactured by Motorolla Corporation. However, it should be understood that the present invention is not limited to any one make of processor and that the invention could be practiced using some other type of a processor, such as a co-processor or an auxiliary processor.

Main Memory 235 contains Operating System 240, XML Parser 250 and Menu Processor 255. Operating System 240 is the operating system known in the industry as PALM OS, which is offered and sold by 3COM Incorporated along with its PALM

III PDA device. As with XML Parser 150, XML Parser 250 can be one of any of the available XML Parsers available for the PALM OS operating system.

LCD Controller 220 is used to render bit-oriented images on the display of Customer Device 200 (i.e., the Auto PC). Serial Port 210 is used to connect Server 200 to other devices such as Wireless Transceiver 260. Wireless Transceiver 260 is used to send and receive wireless transmissions in the form of menus, orders, and other information to and from Server 100 (see Figures 3-5). In particular Wireless Transceiver 260 is capable of sending and receiving transmissions using the Bluetooth standard discussed above. Wireless Transceiver 260 is also capable of sending wireless transmissions using the IRDA standard discussed above.

Customer Device 200 - AutoPC

The second hardware embodiment of Customer Device 200 is a AutoPC automobile computer device that is marketed by Clarion Corporation. However, it should be understood that any other similarly configured automobile computer could be used. As shown, Customer Device 200 comprises Central Processing Unit (CPU) 205, which is connected to LCD Controller 220 and Memory Card 235. Customer Device 200 has also been enhanced to include Serial Port 210 and Wireless Transceiver 260, which are components that are not customarily available in a standard AutoPC. As shown, Serial Port 210 is also connected to Wireless Transceiver 260.

CPU 205 is a Hitachi SH3 embedded processor. However, it should be understood that the present invention is not limited to any one make or type of imbedded processor.

Main Memory 235 contains Operating System 240, XML Parser 250 and Menu Processor 255. Operating System 240 is the operating system known in the industry

as Windows CE, which is offered and sold by Microsoft Corporation. As with XML Parser 150, XML Parser 250 can be one of any of the available XML Parsers available for the Windows CE operating system. Order Processing Server 255 and its threads, Menu Transmission Thread 257 and Order Processing Thread 259, are explained using Figures 4 and 6-10 and the associated text.

LCD Controller 220 is used to render bit-oriented images on the display of Customer Device 200 (i.e., the PALM III). Serial Port 210 is used to connect Server 200 to other devices such as Wireless Transceiver 260. Wireless Transceiver 260 is used to send and receive wireless transmissions in the form of menus, orders, and other information to and from Server 100 (see Figures 3-5). In particular Wireless Transceiver 260 is capable of sending and receiving transmissions using the Bluetooth standard discussed above. Wireless Transceiver 260 is also capable of sending wireless transmissions using the IRDA standard discussed above.

As a final preliminary matter, it is important to note that while the present invention has been (and will continue to be) described in the context of fully functional servers and customer devices, those skilled in the art will appreciate that the mechanisms of the present invention are capable of being distributed as a program product in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of signal bearing media include: recordable type media, such as floppy disks, hard disk drives, and CD ROMs and transmission type media, such as digital and analog communications links including infrared communication links. It should also be noted that while the mechanisms of the present invention are shown to reside on different computer systems, these mechanisms would likely be distributed as a package on a single instance of signal bearing media.

As stated, Figure 3 shows the steps used to carry out the menu transmission thread of the preferred embodiment. Menu Transmission Thread 157 starts its processing in block 300 and then proceeds to retrieve the date and time of day {block 305}. Menu Transmission Thread 157 then uses the retrieved date and time of day to determine the appropriate menu type {block 310}. As shown there are three types of menus used in preferred embodiment, a Special Menu {block 315}, a Breakfast Menu {block 320}, and a NonBreakfast Menu {block 325}. The appropriate menu is selected based on time of day, available quantities of menu items, and/or the existence of a "special." The date is used to determine whether any Special Menu exists (e.g., because of an advertised promotion) and the time of day is used to determine whether the Breakfast Menu or the NonBreakfast Menu should be used. (The menu structure used in the preferred embodiment is presented in Figures 6 and 7 and the associated text.)

Once determined, the appropriate menu is either generated anew or retrieved from Menu/Price database 145. The meta language used in the preferred embodiment is called Menu Definition Markup Language (MDML), which is an extension of the well known eXtensible Markup Language (XML). The document entitled *Extensible Markup Language (XML)*, February 10, 1998, which is the most current specification for XML available at the time of filing, is hereby incorporated by reference. Figure 6, which shows an example MDML menu, is used later in this patent to further explain the benefits and advantages of the present invention as illustrated by the preferred embodiment.

After Menu Transmission Thread 157 retrieves or generates the MDML menu, Menu Transmission Thread 157 transmits the MDML menu in block 335. Menu Transmission Thread 157 then determines if generation or retrieval of a new menu type is necessary {block 340}. This is done by again checking the date and time of day and by determining whether supplies of a particular menu item have been

exhausted (not shown). If a new menu is not needed, Menu Transmission Thread 157 transmits the MDML menu again {block 335}, and again determines whether a new menu is needed {block 340}. Menu Transmission Thread 157 continues to loop in this manner until it determines that a new menu type is necessary {block 340}. When Menu Transmission Thread 157 determines that a new type of menu is necessary, it proceeds to make the selection and generate or retrieve the MDML menu as has been described in the text accompanying blocks 310 through 330.

Figure 4 shows the steps used to carry out the processing of Menu Processor 255 of the preferred embodiment. As described above, Customer Device 200 receives transmission of the MDML menu {see block 335 of Figure 3} when it comes within range of the signal transmitted by Wireless Transceiver 160. This occurs in block 400. Menu Processor 255 then parses the MDML menu using XML parser 250 {block 405}. Once parsed, the MDML menu is formatted and displayed to the user in the manner applicable to Customer Device 200 {block 410}. Menu Processor 255 then waits for the user's menu selections in block 415. Menu selections are made in the manner applicable to Customer Device 200. If a user abort request is received, Menu Processor 255 terminates execution in block 445.

If a user abort request is not received, Menu Processor 255 creates an MDML order document based on the user's selections {block 430} and transmits the order document to Server 100 {block 435}. Figure 10, which shows an example MDML order document, is used later in this patent to further explain the benefits and advantages of the present invention as illustrated by the preferred embodiment. It should be noted that payment information and vehicle identification information are included in the order document of the preferred embodiment. In the preferred embodiment , credit card information is used as payment information and a randomly generated key along with the year, color, make (e.g., Ford), and type (e.g., truck) of vehicle are used as vehicle identification information. Those skilled in the art will

appreciate that other techniques are equally applicable without loss of generality. For example payment information could specify that cash would be used at the time of pick up to pay for the order or could include an account number. Similarly, vehicle identification could be or include license and registration information or be information that identifies Customer Device 200.

After transmission of the order document, Menu Processor 255 waits for a reply from Server 100 {block 440}. When a reply is received, Menu Processor 255 determines whether the selections have been accepted or rejected {block 455}. (As will be described later in the discussion of Figure 5, Menu Processor 255 is able to identify the correct reply based on returned vehicle identification information.) If the user's selections are rejected by Server 100, Menu Processor 255 displays the reasons for the rejection (as transmitted by Server 100) to the user {block 450}, and repeats display of the menu to the user {block 410}. Processing of blocks 415 through 440 then repeats as was described above. If the user's selections are accepted by Server 100, Menu Processor 255 displays the acceptance information (as transmitted by Server 100) to the user {block 460}, and terminates execution in block 445. In the preferred embodiment, acceptance information is information that instructs the user on how to pick up their order; however, other information could be transmitted.

Figure 5 shows the steps used to carry out the processing of Order Processing Thread 159 of the preferred embodiment. Order documents transmitted by Menu Processor 255, are received by Order Processing Thread 159 in block 500. Order Processing Thread 159 parses the received MDML order document using XML parser 150 {block 505}. Menu Processor 255 then analyzes the order and payment information contained in the order document {block 510} to determine whether the order as a whole is valid. An order may be invalid because payment information cannot be verified or because supplies of an ordered item have run out after menu transmission, or for some other reason. If an order is invalid {block 520}, Order

Processing Thread 159 transmits the rejection information to the appropriate Customer Device 200 {block 515}. Note that the vehicle identification information contained in the order document is returned with the rejection so that the respective Customer Device 200s can determine whether a rejection notification is intended for their user. If an order is valid {block 520}, Order Processing Thread 159 transmits the acceptance information in block 525. Again, the vehicle identification information is used to ensure that the acceptance information is processed by the correct Customer Device 200. After transmitting the acceptance information, Order Processing Thread 159 creates a vehicle description from the vehicle identification information included in the order {block 530}, displays the vehicle description along with the order to the restaurant staff {block 540} and terminates execution in block 535.

Figure 6 is a hierarchical diagram showing a portion of the menu structure of the preferred embodiment. Those skilled in the art will appreciate that other menu structures are possible within the spirit and scope of the present invention. Date and Time 600 are used in the manner of a three-way switch to determine the appropriate Meal Type 605. There are three meal types: Special 610, NonBreakfast 615, and Breakfast 620. Under Breakfast 620 there is provision for Meal Category 625, for which there are two possibilities, Food 630 and Drink 645. Beneath Food category 630, there is provision for Item Type 635, and beneath Item Type 635, there is provision for individual items. An example of an item type would be "sandwich" and an example of an item would be "sausage-egg sandwich." Included within Item are the name, size, cost, and quantity of the item. (Item quantity is present in the menu structure for order documents, not for initial presentation of a menu.)

Figure 7 is a hierarchical diagram showing the remaining portion of the menu structure of the preferred embodiment. In particular, Figure 7 shows the NonBreakfast meal type in greater detail. Please refer to the discussion of Figure 6 for details on the depicted structure.

Those skilled in the art understand that the XML language is extended through the use of what is referred to as a document type definition (DTD for short). A DTD is basically a set of definitions that provide information to an XML parser about how individual tags within the definitions are to be handled. One or more DTDs can be used to extend the capabilities of XML. In essence, the addition of each DTD to the base XML language creates a new tag-oriented language. As has been mentioned earlier, the mechanisms of the preferred embodiment utilize an extension to the XML language called the Menu Definition Markup Language (MDML).

Figure 8 is a diagram showing the DTD for the MDML language. The DTD for MDML, named Menu.dtd, includes MENU element 800, which includes the parameters TIMESTAMP and MEALTYPE+; ORDER element 801, which includes the parameters TIMESTAMP, VEHID, PYMT, and MEALCAT*, VEHID element 802, which is parsed by Order Processing Thread 159 and Menu Processor 255; PYMT element 803, which is parsed by Order Processing Thread 159; TIMESTAMP element 805, which includes the parameters DATE and TIME; DATE element 807, which is parsed by Menu Processor 255 and Order Processing Thread 159; TIME element 809, which is parsed by Menu Processor 255 and Order Processing Thread 159; MEALTYPE element 811, which includes the MEALCAT+ parameter; MEALTYPE attribute list 813, which includes the required CDATA parameter "name;" MEALCAT element 815, which includes the ITEMTYPE* and ITEM* parameters; MEALCAT attribute list 817, which includes the required CDATA parameter "category;" ITEMTYPE element 819, which includes the ITEM+ parameter; ITEMTYPE attribute list 821, which includes the required CDATA parameter "type;" ITEM element 823, which does not include a parameter; ITEM attribute list 825, which includes the required CDATA parameters "type," "size," and "price," and the implied CDATA parameter "quantity."

Figures 9 and 10 are example MDML documents that are used in the Detailed Description to help explain the innerworkings of the preferred embodiment. Figure 9 shows one example breakfast menu that conforms to the menu structure of the preferred embodiment. Breakfast menu 900 is one example of a MDML menu that could be generated/retrieved in block 330 of Figure 3 and transmitted in block 335 of Figure 3.

As with all well formed XML documents, breakfast menu 900 begins with the appropriate XML version statement (statement 902). Next, breakfast menu 900 includes DOCTYPE statement 905, which identifies the correct DTD to the XML parser (XML Parser 250 in this case). Again, the DTD contains the rules (called the grammar) which are used as the basis for parsing the document. What follows next is a series of tags. The reader should note that each tag (e.g., MENU 910 and TIMESTAMP 915) is somehow defined in the DTD for MDML (i.e., menu.dtd of Figure 8).

MENU tag 910 signals the beginning of a menu. TIMESTAMP tag 915 denotes the beginning of a timestamp. A timestamp, as defined in the DTD, includes two fields, a date field and a time field. In menu 900 the date is set to be January 28, 1999, and the time is set to be 7:53:22. MEALTYPE tag 925 is next, showing that the meal type is "breakfast." MEALCAT tags 930 and 942 are used to specify the two types of meal categories (i.e., FOOD and DRINK). Then, within each meal category, there is specified different item types (see for example ITEMTYPE tags 935 and 941). At the finest level of granularity the items themselves are specified. See for example ITEM tag grouping 940, where the different types of sandwiches are specified along with the cost of each. Note here that even though quantity is specified as a parameter in the menu structure definition of ITEM on Figures 6 and 7 is considered optional, given that it has no relevance in the context of the presentation of a menu. Note also that technically each tag construct is made up of start-end tag pairs

(see for example end tag grouping 950, which match up with some of the earlier-explained start tags).

Figure 10 shows one example of an order that conforms to the menu structure of the preferred embodiment. Order 1000 is one example of an MDML order that could be created in block 430 of Figure 4 and transmitted in block 435 of Figure 4.

Again as with all well formed XML documents, order 1000 begins with the appropriate XML version statement (statement 1002). Next, order 1000 includes DOCTYPE statement 1005, which identifies the correct DTD to the XML parser (XML Parser 150 in this case). The DOCTYPE statement is followed by ORDER tag 1007, which signals the beginning of an MDML order. TIMESTAMP tag 1010 serves the same purpose as described above. VEHID tag 1015 is used to uniquely identify a particular vehicle. For example, in this particular order the vehicle is a 1997, tan, Ford truck and the random key generated by Customer Device 200 is the number 2175. PYMT tag 1017 is used to identify the form of payment chosen by the customer. Here the data associated with PYMT tag 1017 indicates that the customer wants to pay for this order with a credit card. Another option would be for the customer to indicate cash as a preference and pay for the order at the time the order is picked up. Following PYMT tag 1017 is the actual order itself. Here the customer is ordering two Sausage-Egg sandwiches, a small Hashbrowns, and a milk (see ITEM tags 1020, 1025, and 1030).

Physical Environment - Existing Systems

For the purposes of compatibility, the mechanisms of the preferred embodiment can function in conjunction with existing drive-up window technology or in lieu of such technology. Customers who do not have a device capable of functioning as Customer Device 200 or who simply prefer to use an existing two-way speaker

configuration are not precluded from doing so by the mechanisms of the preferred embodiment. Indeed, the display used to display vehicle identification and order information (see block 540 of Figure 5) would simply display information regarding both types of orders (i.e., electronic and nonelectronic) to the restaurant staff. In another embodiment of the present invention, the transmitter of Server 100 could be located at a first location and the receiver of Server 100 could be located at a second location , perhaps at an existing order station (i.e., the location of the existing two-way speaker), such that orders would naturally be queued in a manner similar to that of current drive-up window configurations. While this latter embodiment would be less general, it would eliminate the complexity associated with ensuring that the correct Customer Device 200 processes transmitted rejection and acceptance information. Another possible embodiment of the present invention is the use of Customer Device 200 to transmit orders as described above, but to have the order filled at a walk-up station or window instead of at a drive-up window as is contemplated in the primary embodiment. In this latter embodiment, the acceptance information transmitted by Server 100 (see block 525 of Figure 5) would comprise an order number that would be used when picking up one's order.

The embodiments and examples set forth herein were presented in order to best explain the present invention and its practical application and to thereby enable those skilled in the art to make and use the invention. However, those skilled in the art will recognize that the foregoing description and examples have been presented for the purposes of illustration and example only. The description as set forth is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching without departing from the spirit and scope of the following claims.